# Bandwidth Adaptation Algorithms for Adaptive Multimedia Services in Mobile Cellular Networks

Taekyoung Kwon and Yanghee Choi
Seoul National University
Seoul, Korea
{tkkwon,yhchoi}@mmlab.snu.ac.kr

Sajal K. Das
University of Texas at Arlington
Arlington, TX 76019
das@cse.uta.edu

**Abstract**

*The fluctuation of available link bandwidth in mobile cellular networks motivates the study of adaptive multimedia services, where the bandwidth of an ongoing multimedia call can be dynamically adjusted. We analyze the diverse objectives of the adaptive multimedia framework and propose two bandwidth adaptation algorithms (BAAs) that can satisfy these objectives. The first algorithm, BAA-RA, takes into consideration revenue and "anti-adaptation" where anti-adaptation means that a user feels uncomfortable whenever the bandwidth of the user's call is changed. This algorithm achieves near-optimal total revenue with much less complexity compared to an optimal BAA. The second algorithm, BAA-RF, considers revenue and fairness, and aims at the maximum revenue generation while satisfying the fairness constraint defined herein. Comprehensive simulation experiments show that the difference of the total revenue of BAA-RA and that of an optimal BAA is negligible. Also, numerical results reveal that there is a conflicting relationship between anti-adaptation and fairness.*

## 1 Introduction

Due to the rapid advances in wireless/mobile communications technology, there have been tremendous efforts in deploying multimedia services in this environment. However, the link bandwidth of mobile cellular networks is still a bottleneck. The scarcity in wireless resources motivates us to research on the adaptive multimedia services which can operate over a wide range of available

bandwidth [1, 2, 3, 4, 5].

The adaptive multimedia paradigm can play an important role to mitigate the highly-varying resource availability in wireless/mobile networks. For example, in a non-adaptive multimedia framework where the bandwidth of a call is fixed, the handoff call is forced to terminate if there is no available bandwidth in the forward cell. In contrast, in an adaptive multimedia framework, it is possible to overcome the link overload condition by reducing the bandwidth of individual calls, which we call *bandwidth adaptation*, thereby accepting the handoff call.

Originally, the concept of adaptive multimedia service was introduced in wired networks to cope with network congestion. Broadly, two approaches in adaptive multimedia have been proposed in the literature. In the first approach, the source adjusts the *rate* (or *bandwidth*) of a multimedia stream depending on the condition of the network [6, 7]. The value of the rate is usually continuous. In the second approach, on the other hand, the multimedia stream is compressed in the form of *layered* (or *hierarchical) coding* to support heterogeneous receivers. Thus, each receiver can selectively choose the subset of layered coding depending on both its capability and bandwidth availability [8, 9]. Another scheme in the second approach is transcoding where one multimedia coding is changed into another (e.g., MPEG-1,2 into H.263) [1, 10].

More recently, some *bandwidth adaptation algorithms* (BAAs) have been proposed in wireless/mobile networks [2, 3, 4]. Bharghavan et al. [2] seek to achieve optimal bandwidth allocation over the whole network. However, the message overhead is inherently high for this scheme which also assumes continuous values of bandwidth. A more generalized BAA is proposed by Talukdar et al. [3] who investigate the tradeoff between network overhead and optimal bandwidth allocation. While the algorithms in [2, 3] aim at optimal bandwidth adaptation over the whole network, Das and Sen [4] propose a BAA from a cell's perspective. They exploit the tradeoff between the carried traffic and bandwidth degradation, thus deriving an optimal policy to maximize total revenue.

In this paper, we adopt the layered coding approach where the bandwidth of a call can take a set of discrete values, as in [1, 3, 12] and a sender transmits the layered coding of a multimedia stream to a mobile terminal (MT). If a cell is "underloaded," the MTs (with an ongoing adaptive multimedia call) in the cell receive the full multimedia stream, i.e., the whole set of layered coding. However, if congestion takes place in the cell, the layered coding is adapted at the base station or at the adaptation-enabled switch. In other words, the subset of the layered coding is *filtered* or *transcoded* at the base station to adapt to the situation of the "overloaded" cell [1, 13].

Additionally, as suggested in [1, 4, 12], we focus on the bandwidth of individual cells rather than the whole network [2, 3], due to the following reasons:
(1) We believe that the bandwidth adaptation in the whole network is of high-complexity and is likely to have a negative impact on the data traffic (e.g., TCP's congestion control).
(2) Without loss of generality, the wireless link bandwidth is much more valuable than the bandwidth of wired networks. As a result, wireless bandwidth will be a dominant factor in overall connection from a revenue point of view.
(3) We believe that the probability of overload (when the bandwidth adaptation is required) in a cell is presumably bounded by a call admission control (CAC) algorithm (e.g., [4, 12]).

With the help of a CAC algorithm, we propose bandwidth adaptation algorithms (BAAs) for the case in which a cell is overloaded only occasionally. Compared to the fixed networks, the fluctuation in resource availability in wireless/mobile networks results from two inherent features: fading and mobility. The fading in a wireless channel is highly varying with temporal and spatial dependencies and interferences. We assume that the effect of fading can be mitigated by a rich-function transmission/reception wireless subsystem (e.g., [14]). Hence, our adaptive multimedia framework takes into consideration only mobility (or equivalently, handoff). That is, an adaptive multimedia call may have to change its bandwidth when there is a new call arrival, a call completion, or an incoming/outgoing handoff. Therefore, a BAA should choose which call(s) to adapt and how much bandwidth of each chosen call is changed.

In this paper, we propose two BAAs considering the diverse quality-of-service (QoS) requirements in the adaptive bandwidth framework. The first algorithm, called BAA-RA, considers the *revenue* and the *adaptation cost* (the penalty for each bandwidth change of a call). Whereas, the second algorithm, called BAA-RF, takes into account the revenue and the *fairness* property.

The paper is organized as follows. The diverse objectives of the bandwidth adaptation framework are identified in Section 2. The model of adaptive multimedia stream is described in Section 3. Our BAAs are proposed in Section 4 while Section 5 evaluates our algorithms in terms of defined objectives through simulation experiments. Then the concluding remarks are given. The analysis of the CAC algorithm under consideration is provided in appendix.

## 2 BAA Objectives

The objectives or requirements of BAAs can be categorized from the standpoint of service providers as well as service users. For example, service providers of adaptive multimedia may need to satisfy the following two requirements.

- *Revenue*

  The primary concern is the revenue. The revenue rate of an ongoing call is mainly dependent on the current bandwidth of the call. In this paper, we assume that the revenue rate of a call is determined by a monotonically increasing revenue function of the bandwidth. Figure 1 shows two types of revenue functions, namely linear and convex.

- *Complexity*

  Another concern of service providers is the cost of the network equipment (e.g., base station). The implementation complexity of deploying the adaptive multimedia services has a significant effect on the cost of the deployment. Therefore, the overall complexity of a BAA should be reasonable (say, polynomial-time complexity).

On the other hand, the service users may have the following (QoS) requirements in terms of bandwidth. In our adaptive multimedia framework, QoS parameters such as delay and jitter are not taken into account.

- *Quality*

  Obviously, the more the bandwidth allocated for a call, the more satisfied is a service user. We assume that the perceived quality of an adaptive multimedia stream is dependent only on the currently allocated bandwidth of the call. Furthermore, the quality function of bandwidth is the same as the revenue function considered. Thus, maximizing revenue implies maximizing the perceived quality of users.

- *Anti-adaptation*

  A service user may feel uncomfortable whenever the adaptation (i.e., the change of bandwidth of the user's call) occurs as proposed in [2, 3, 12]. We denote this QoS requirement as "anti-adaptation" and consider the following three formulations for anti-adaptation.

  (i) *negative revenue*: Each adaptation has a penalty associated with it [2, 4]. Accordingly,

BAAs should take into account both positive revenue for the bandwidth of a call and negative revenue for the change of bandwidth of the call.

(ii) *adaptation rate*: The upper bound on the number of adaptations per unit time is guaranteed in this formulation. A proposed BAA keeps track of the information about adaptation rate of each call and adapts the bandwidth of calls whose adaptation rate is less than the specified upper bound. (The BAA in [3] also seeks to achieve fairness in adaptation rate.)

(iii) *inter-adaptation time*: This formulation can enforce that the time between two successive adaptations of a call is larger than some QoS value. This parameter can be regarded as a more stringent QoS requirement than (ii).

- *Fairness*

  Another important aspect considered by the service users in a cell is fairness. If the bandwidth values of adaptive multimedia services are assumed to be continuous, then fairness would be ensured because every call would need to reduce its bandwidth by the same ratio, which makes the resulting sum of bandwidth of all calls equal to the capacity of a cell. However, the bandwidth values of adaptive multimedia under consideration being discrete, there exists a problem of fairness between calls. More precisely, there are two kinds of fairness: inter-class fairness and intra-class fairness. *Inter-class fairness* means that the bandwidth of a cell should be fairly partitioned among service classes. Whereas, *intra-class fairness* means that each call in a class should be allocated an even portion of the bandwidth partition for that class.

It is not difficult to see that there is a conflicting relationship between anti-adaptation and fairness criteria. Intuitively, fairness tries to adapt as many calls as possible by an equal ratio. On the contrary, anti-adaptation attempts to reduce the number of calls whose bandwidth is to be changed. Therefore, we believe that a BAA which seeks to satisfy anti-adaptation and fairness at the same time is not meaningful. To this end, we propose two algorithms: BAA-RA for revenue and anti-adaptation, and BAA-RF for revenue and fairness.

# 3 Model Description

We consider a general model consisting of $K$ classes of adaptive multimedia calls in wireless/mobile cellular networks. This section describes our traffic model, adaptive multimedia model, and a CAC

algorithm.

### 3.1 Traffic Model

For simplification, we consider a single isolated cell. Call arrivals of class-$i$, where $i = 1, 2, ..., K$, are assumed to form a Poisson process with mean arrival rate $\lambda_i$. Also, handoff calls of class-$i$ are assumed to arrive as a Poisson process with mean rate $h_i$. The call holding time (CHT) of a class-$i$ call is assumed to follow an exponential distribution with mean $1/\mu_i$. For the mobility characterization, we assume that the cell residence time (CRT), i.e., the amount of time during which a call stays in a cell before handoff, follows an exponential distribution with mean $1/h$. We assume that the CRT is independent of class; that is, calls in any class follow the same CRT distribution. Note that $h$ represents the mean handoff rate.

### 3.2 Model of Adaptive Multimedia

According to the adaptive multimedia paradigm, a multimedia call can dynamically change its bandwidth during its lifetime. Assume that the bandwidth of a class-$i$ call takes its value from the set $V_i = \{b_{i,1}, b_{i,2}, ..., b_{i,s_i}\}$ where $b_{i,j} < b_{i,j+1}$ for $j = 1, ..., s_i - 1$. Here, $s_i$ is the number of possible bandwidth values that a class-$i$ call can be allocated.

Figure 2 illustrates an example of an adaptive multimedia stream of a class-1 call where $s_1 = 3$. For the purposes of illustration, let us assume that a cell has only class-1 calls. If the cell is underloaded, every call in the cell will be allocated its maximum bandwidth, $b_{1,3}$. Otherwise, depending on how much the cell is overloaded, one or more calls in the cell will be allocated $b_{1,2}$ or $b_{1,1}$. The difference $\Delta b_{1,j} = b_{1,j+1} - b_{1,j}$, for $j = 1$ and 2, is the $j$-th segment of multimedia stream with respect to the minimum bandwidth ($b_{1,1}$) of the class-1 call.

As mentioned earlier, the revenue rate for a call is dependent only on the bandwidth allocated to that call. Henceforth, the *revenue rate* will simply be referred to as the *revenue*. A revenue function for a class-$i$ call will be denoted by $r_i(\cdot)$. For example, when bandwidth $b_{i,j}$ $(j = 1, ..., s_i)$ is allocated to a class-$i$ call in progress, the corresponding revenue is $r_i(b_{i,j})$.

Assuming a fixed channel allocation (FCA) scheme, the total bandwidth (number of channels) in each cell is the same, denoted by $C$. Furthermore, we assume that the bandwidth of a wideband call can be scattered across the bandwidth pool as in [15, 18].

### 3.3 Our CAC Algorithm

The current state of a cell is represented by a vector $X^T = (x_1, x_2, ..., x_K)$, where $x_i$ denotes the number of ongoing class-$i$ calls in the cell. Let $N_X$ be the total number of calls in state $X$. Then, the role of a BAA is to allocate/reallocate the appropriate bandwidth to individual $N_X$ calls satisfying the previously mentioned objectives whenever the cell is overloaded. The probability that the cell is overloaded is determined by a CAC algorithm, which decides the state space of a cell. We apply a simple CAC algorithm called the upper limit (UL) CAC algorithm [17] which is also classified as the coordinate-convex CAC [18].

To adopt the UL CAC algorithm in our context, we take into account the *cell overload probability*, $P_O$, which is defined as the sum of the steady probabilities for the states where at least one call in the cell should be "degraded" in bandwidth. (A cell is designated as "overloaded" if at least one call in the cell needs to be degraded.) This degradation happens when the bandwidth of the cell is not enough to accommodate every call with its maximum bandwidth. That is, $b_{max} \cdot X > C$ where $b_{max} = (b_{1,s_1}, b_{2,s_2}, ..., b_{K,s_K})$ is a vector of maximum bandwidth values for all classes.

In this CAC algorithm, a newly arriving call is blocked if the number of class-$i$ calls is greater than or equal to a *threshold* value, say $t_i$. Whereas, the handoff incoming call will not be blocked because the forced termination of a handoff call is much more unbearable to users than blocking of a new call. By adjusting the value of $t_i$ for each class, we can derive a CAC algorithm which satisfies the QoS requirements (e.g., upper bound on $P_O$).

## 4 Proposed Bandwidth Adaptation Algorithms

In this section, we propose two bandwidth adaptation algorithms (BAAs) which meet the objectives discussed in Section 2. As mentioned earlier, the revenue for the bandwidth is assumed to be equal to the perceived quality by a service user for the same bandwidth. In other words, we need not consider the perceived quality separately from the revenue. The following presents a BAA for revenue and anti-adaptation, and another BAA for revenue and fairness. Before proceeding further, let us first present a basic BAA which considers only revenue in a cell.

### 4.1 BAA for Revenue Only (BAA-R)

A naive implementation of BAA for optimal revenue is NP-hard and has exponential time complexity [16]. Hence, the proposed BAA-R seeks to achieve a near-optimal revenue with polynomial time complexity in an overloaded cell. It is based on a greedy approach [19], featuring $d$-combination lookahead.

#### 4.1.1 Graph Abstraction

The BAA-R uses the following graph formulation. The "start" node is set up initially. Then, degradable bandwidth segments (see Figure 2) of each ongoing call correspond to the nodes laid out in a chain as illustrated in Figure 3, which contains two class-1 calls and one class-2 call. Thus the number of chains is the same as the number of calls in progress in a cell. In each chain, the first degradable segment of each call is connected to the "start" node. Bear in mind that the first degradable segment of a class-$i$ call is $\Delta b_{i,s_i-1} = b_{i,s_i} - b_{i,s_i-1}$. Similarly, the second degradable segment, of the call is connected to the first degradable segment and so on. In Figure 3, the direction of the edges are downwards which reflects the degradation order. Furthermore, the minimum bandwidth segment, $b_{i,1}$, of each call is not represented in the graph because it is not degradable.

The cost of the edge to the node $\Delta b_{i,j}$ is expressed by $c_{i,j}$ which is calculated by dividing the revenue of the node by its bandwidth size. The revenue of the node denotes the decreasing revenue by degrading the bandwidth segment (represented by the node) of the call. Thus, the lower the cost of the edge, the more bandwidth can be degraded with the same revenue loss by choosing the edge. The cost $c_{i,j}$ is calculated as

$$c_{i,j} = \frac{r_i(b_{i,j+1}) - r_i(b_{i,j})}{\Delta b_{i,j}},\ j = 1, 2, ..., s_i - 1 \tag{1}$$

For the convenience of notation, the graph in Figure 3 can be modified as follows. The "start" node is denoted by node 0. Each degradable bandwidth segment of each call is indexed as illustrated in Figure 4. Here, the index of the node can be ordered arbitrarily. Also, each edge to a node $n$ is associated with a two-tuple $(c(n), b(n))$ where $c(n)$ is the cost of the edge to node $n$ and $b(n)$ is the bandwidth size of the node $n$.

### 4.1.2 Description of BAA-R

The BAA-R scheme stems from Prim's minimum spanning tree (MST) algorithm [19]. The nodes are selected one by one at each step, which corresponds to the selection of the bandwidth segments to be degraded. The main difference between the BAA-R and the Prim's algorithm is that BAA-R spans the tree until the sum of the bandwidths of selected nodes is at least $b_{max} \cdot X - C$. For this purpose, the BAA-R keeps track of the costs and bandwidths of all nodes. The important feature of BAA-R is the "$d$-combination lookahead" method, where $d$ is a positive integer indicating how many combination of edges to be searched to find out a near-optimal bandwidth adaptation. In the algorithms to be proposed, the selection of a node and the selection of an edge to the node have the same meaning. Henceforth, we use the terms *node* and *edge* interchangeably.

When $d = 1$, we look ahead only one combination of the outgoing edges. More exactly, before selecting the edge with the minimum cost at each step, we compare the maximum bandwidth of the edge among $U(N)$ edges with the required bandwidth, say $b_{req}$. Here, $N$ is the set of already explored nodes, and $U(N)$ is the set of unexplored outgoing edges of nodes in the set $N$. Note that the number of edges in $U(N)$ is always the same as the number of ongoing calls. Furthermore, $b_{req} = b_{max} \cdot X - C - sum_b$ is the amount of insufficient bandwidth to terminate the BAA-R algorithm, where $sum_b$ is the sum of bandwidth of the already explored nodes (or equivalently, edges) at that instant. If $b_{req}$ is less than or equal to the maximum bandwidth of the edges in the set $U(N)$, then we investigate all the outgoing edges individually (so-called 1-combination). Among the investigated edges, we choose that edge with the minimum revenue loss while the bandwidth of the edge is greater than $b_{req}$. Otherwise, we choose the edge with the minimum cost and repeat this step (lines 3-8 in Table 1) until $b_{req}$ is less than or equal to the maximum among bandwidth values of the edges in $U(N)$.

When $d = 2$, this algorithm searches more adaptation cases (2-combination). Before selecting the edge with the minimum cost, we compare $b_{req}$ with the sum of the two largest bandwidth values in $U(N)$. If this sum is greater than $b_{req}$, we investigate every combination of two edges in $U(N)$. Among the investigated cases, the minimum revenue loss of two edges is chosen on the condition that the bandwidth sum of two edges is greater than or equal to $b_{req}$.

Table 1 describes our BAA-R scheme. The notation $\sum^d S$ represents the sum of the $d$ largest bandwidth values of the edges in the set $S$. Let $R(S)$ and $B(S)$ be respectively the sum of revenue loss and the sum of bandwidth values of the selected edges in the set $S$. In addition, $R_{min}$ is

the minimum revenue loss by degrading bandwidth segments in the set $S_{min}$. After the BAA-R is executed, the nodes in $N$ and $S_{min}$ will be degraded. For example, if the nodes within the dotted line in Figure 4 are selected by the BAA-R procedure, then two class-1 calls are allocated a bandwidth $b_{1,s_1-1}$ and a class-2 call is allocated a bandwidth $b_{2,s_2-2}$.

Note that the combination of less than $d$ edges is also searched (line 10 in Table 1). Clearly, as $d$ increases, the corresponding revenue increases because the algorithm searches more cases of bandwidth adaptation.

### 4.2 Revenue and Anti-adaptation (BAA-RA) Algorithm

The scheme BAA-RA considers both revenue and anti-adaptation. It is equivalent to the BAA-R except that the notion of anti-adaptation is incorporated into the graph. Among the three formulations of anti-adaptation (see Section 2), we choose the negative revenue because the adaptation rate and inter-adaptation time can be easily incorporated with other objectives. That is, we just need to exclude from BAAs the calls which will violate the requirements of adaptation rate or inter-adaptation time.

In general, the adaptation cost may be dependent on the characteristics of the application service. For example, the adaptation costs for audio, video, and data services may be different from each other. Accordingly, the penalty for anti-adaptation is modeled by $a_i$ (a positive value) for each class-$i$ call and will be subtracted while calculating the total revenue.

This penalty $a_i$ is incorporated into the graph for BAA-R algorithm (Figure 4) as follows. Note that $a_i$ is the penalty value at the instant of adaptation, whereas the revenue is the notion of rate, i.e., a user pays the revenue for an allocated bandwidth during which the user is serviced by that bandwidth. Hence, we formulate the "adaptation rate" instead of "adaptation cost" as follows. First, the time, $T_X$, from state $X$ to any next state is estimated. Then, $a_i$ is divided by $T_X$ to make the dimensionality equal (revenue rate and adaptation rate). From state $X$, the expected time until the next state will be given by the reciprocal of the sum of state transition rates, and is given by

$$T_X = \left[ \sum_{i=1}^{K} \{\lambda_i I(x_i < t_i) + h_i + (\mu_i + h)x_i\} \right]^{-1} \tag{2}$$

where $I(\cdot)$ is an indicator function which equals 1 if its argument is true and equals 0 otherwise. Recall that $x_i$ is the current number of class-$i$ calls; $\lambda_i$ and $h_i$ are respectively the arrival rates of

new and handoff class-$i$ calls; Also, $\mu_i$ and $h$ are the service rate and handoff rate, and $t_i$ is the threshold denoting the blocking of a cell.

With this formulation of adaptation cost, if the bandwidth of a call before adaptation is less than its maximum bandwidth, we modify the cost of edges as in Figure 5. Here $r(n)$ and $b(n)$ respectively denote the revenue loss and the bandwidth amount of the node $n$. When an edge is selected while running the BAA-RA scheme and the resultant bandwidth of the call is equal to the bandwidth of the call before adaptation, the revenue loss of the edge is decreased by $a_i/T_X$. Subsequently, the cost of the edge is also decreased. (We implicitly assume that the decreased cost is a non-negative value.) This formulation may encourage the selection of the edge which makes the bandwidth of the call remain unchanged (see c(1) in Figure 5(a)), thereby avoiding the penalty of the adaptation cost. The cost of an edge does not necessarily mean the revenue loss directly; it just indicates how much revenue loss is incurred per unit of degraded bandwidth.

On the contrary, when the bandwidth of a call is more degraded than the previous bandwidth value, the selection of the edge is discouraged by increasing the revenue loss by the amount $a_i/T_X$ (see $c(4)$ in Figure 5(a)). Note that there is no change in the costs of other edges. The bold line of the multimedia stream in Figure 5(a) indicates the allocated bandwidth of the call before this adaptation.

On the other hand, in the case that the maximum bandwidth is allocated to a call before adaptation, the formulation of adaptation cost for the call is simpler as depicted in Figure 5(b). Again, the bold line represents that the call is allocated its maximum bandwidth before adaptation. If a new call is accepted and an adaptation has to be invoked, the call is represented in the graph as shown in Figure 4. In addition, a handoff incoming call has its previously allocated bandwidth. Thus, the handoff call can be represented in the graph as depicted in either Figure 5(a) or Figure 5(b).

The same algorithm in Table 1 applies to this modified graph. Depending on how adaptation cost is assigned for each class of call, there will be some skewness in adaptation. For example, calls of the adaptive multimedia service whose $a_i$ is relatively low, may be most frequently adapted when the cell is overloaded.

### 4.3 Revenue and Fairness (BAA-RF) Algorithm

The BAA-RF scheme seeks to find out the best bandwidth allocation (not necessarily optimal from the revenue point of view) while satisfying fairness. We define the *fairness* from the standpoint of discrete bandwidth adaptation and discuss how to maintain the fairness hereafter. For the convenience of explanation, we compare the fairness of discrete bandwidth adaptation with that of continuous bandwidth adaptation.

In continuous bandwidth adaptation, an overloaded cell satisfies $b_{max} \cdot X > C$. So every call only needs to be allocated its maximum bandwidth value multiplied by the ratio $\frac{C}{b_{max} \cdot X}$ where for state $X$, the bandwidth partition $B_i$ for all class-$i$ calls will be given by

$$B_i = \left( \frac{b_{i,s_i} x_i}{b_{max} \cdot X} \right) C \tag{3}$$

This equation represents the "inter-class fairness." From $B_i$, the allocated bandwidth to each class-$i$ call will be given as $\frac{B_i}{x_i}$ which defines the "intra-class fairness."

Let us first try to maintain intra-class fairness of discrete bandwidth adaptation. Let $f_i = \frac{B_i}{x_i}$ be the target bandwidth value for perfect intra-class fairness. If $f_i \in V_i$, i.e., it equals some discrete bandwidth value of class-$i$, then we need to allocate that value to every class-$i$ call. (Recall that $V_i$ is the set of possible bandwidth values of a class-$i$ call.) However, it is hard to expect such an ideal case; accordingly, we assume two bounds $l_i$,$g_i \in V_i$ on bandwidth values such that $l_i < f_i < g_i$. In this paper, the intra-class fairness will be satisfied if every class-$i$ call is allocated a bandwidth of either $l_i$ or $g_i$.

Let us now explain how inter-class fairness is maintained in discrete bandwidth adaptation. Among $x_i$ class-$i$ calls, suppose that $n_{l_i}$ calls will be allocated a bandwidth $l_i$ while the remaining $x_i - n_{l_i}$ calls will be allocated $g_i$ to meet intra-class fairness. For the best inter-class fairness, we aim to derive the value of $n_{l_i}$ which makes the real bandwidth sum of class-$i$ calls close to $B_i$. Alternatively, $n_{l_i}$ can be calculated by considering that the weighted average of the bandwidth values of $x_i$ calls should be close to $f_i$. More precisely, the number of calls with bandwidth $l_i$ can take two candidate values $y_i$ and $y_i + 1$ for class-$i$ to meet both inter-class fairness and intra-class fairness. This implies

$$\frac{(y_i + 1)l_i + (x_i - y_i - 1)g_i}{x_i} < f_i < \frac{y_i l_i + (x_i - y_i)g_i}{x_i}. \tag{4}$$

Depending on the number of calls with bandwidth $l_i$, the real sum of bandwidth values, say

$R_i$, of all class-$i$ calls may be greater or smaller than $B_i$. In this paper, inter-class fairness will be maintained if $|R_i - B_i| < g_i - l_i$. If either of the two weighted averages in Eq. (4) equals $f_i$ coincidentally, then there is only one adaptation case for that class.

Finally, for both kinds of fairness, there are at most two alternatives for each class. It is expected that the number ($K$) of adaptive multimedia service classes is fairly small because the main adaptive multimedia is focused on video stream [1, 6, 7, 8, 9]. As a result, we can search all possible cases of bandwidth adaptation with time complexity $O(2^K)$. Among these cases satisfying the best fairness, we need to choose only the one with the maximum revenue such that the sum of bandwidth values is less than or equal to $C$.

# 5 Simulation Experiments

According to our model described in Section 3, we simulate one cell to evaluate the performance of the proposed BAAs with various objectives. First, we discuss how to quantify the objectives of BAAs and also introduce two other adaptation schemes for comparison purposes. Next, the model of adaptive multimedia services used in the simulation experiments is described. Finally, numerical results are presented under diverse simulation scenarios.

## 5.1 Performance Metrics

We use three performance metrics: (i) total revenue including revenue and adaptation cost, (ii) adaptation cost only, and (iii) fairness (intra-class and inter-class). Total revenue is calculated by subtracting the adaptation costs from the sum of revenue of individual calls in a cell. Suppose an adaptation occurs at time $t$, and no more adaptation takes place in $(t, t']$. Then, the total revenue, $TR$ (the first metric), generated during the interval $[t, t']$ due to the adaptation at time $t$ is given by

$$TR = (t' - t) \sum_{i=1}^{K} \sum_{j=1}^{x_i} r_i(B(i,j)) - \sum_{i=1}^{K} a_i z_i \tag{5}$$

where $z_i$ is the number of class-$i$ calls whose bandwidth is changed at time $t$ and $B(i,j)$ is the allocated bandwidth to the $j$-th call in class-$i$ at time $t$ by applying a BAA. The second metric, the adaptation cost in a given cell ($AC$), is expressed by

$$AC = \sum_{i=1}^{K} a_i z_i \tag{6}$$

Since fairness is difficult to quantify, both kinds of fairness is defined by the standard deviation from the the exact bandwidth value (in continuous bandwidth adaptation case). Intra-class fairness (IF) and inter-class (XF) fairness for state $X$ during $[t, t']$ is given by Eqs. (7) and (8), respectively.

$$IF = (t' - t)\sqrt{\sum_{i=1}^{K}\sum_{j=1}^{x_i}(B(i,j) - f_i)^2} \tag{7}$$

$$XF = (t' - t)\sqrt{\sum_{i=1}^{K}(R_i - B_i)^2} \tag{8}$$

For the purpose of comparison with BAA-RA and BAA-RF, we introduce two more schemes. The first algorithm is a bandwidth adaptation algorithm for optimal total revenue (BAA-O) whose objective is the same as BAA-RA. The naive implementation of BAA-O as a reference algorithm, may investigate every adaptation case (every bandwidth for every call), and thus achieving the maximum total revenue for every state $X$. As a result, it has an exponential complexity $O(s_M^{N_M})$, where $N_M$ is the maximum among possible $N_X$ values and $s_M$ is the maximum among $s_i$ for all $i$. In contrast, the BAA-RA scheme has polynomial time complexity of $O(N_M s_M + {N_M}^d)$ with $d$-combination lookahead.

The other reference algorithm is BAA-A which attempts to minimize only the adaptation cost. Thus, whenever an incoming call (either a newly originating call or a handoff call) is accepted and subsequently an adaptation should be invoked, one or more calls with the lowest adaptation cost will be degraded by the least possible amount of bandwidth to accommodate the incoming call. However, there is no adaptation when a call terminates or hands-off to adjacent cells; that is, the released bandwidth of outgoing calls will be used for future incoming calls.

### 5.2 Simulation Model of Adaptive Multimedia

Considering the current wireless voice coding technology, we can presumably think that the bandwidth value for adaptive voice is less than tens of Kbps. In the case of video compression technology [8, 9], we take into account two classes: low quality video (H.263-based) and high quality video (MPEG-based). We assume that the low quality video takes its rates between tens of Kbps and

hundreds of Kbps, while the high-quality video can be adaptively delivered at rates between hundreds of Kbps and Mbps. Henceforth, these three adaptive services will be denoted as class 1, class 2, and class 3, respectively.

We consider that the setting of bandwidth for adaptive multimedia services can be determined flexibly using adaptation techniques (e.g., filtering and transcoding [1, 13]) and/or compression algorithms (e.g., motion estimation and conditional replenishment [8, 9]). That is, we can adjust $s_i$ and bandwidth values in $V_i$ arbitrarily for each class. Recall that $s_i$ is the number of possible bandwidth values of a class-$i$ call and $b_{s_i}$ is the maximum bandwidth value of a class-$i$ call. Table 2 shows the bandwidth values of adaptive multimedia services used in our simulation experiments. Here, the bandwidth value is expressed by the unit of an 8 Kbps channel because currently low-rate voice coding schemes like QCELP and VSELP deliver speech over a wireless link at a rate of 8 Kbps or less [11]. Even though there may be some discrepancy between the real adaptive services and our simulation model of adaptive multimedia, we believe that our model can reflect the real system's behavior.

As regards to the revenue generated by the bandwidth values, we consider two functions: the linear revenue function $r_i(x) = x$, and the convex revenue function $r_i(x) = \frac{-(x-b_{s_i})^2+{b_{s_i}}^2}{b_{s_i}}$. (The convex revenue function is widely accepted in the literature [2, 4].) Moreover, in order to observe the impact of adaptation cost ($a_i$), we assume that this cost for each of the three multimedia classes is proportional to the corresponding maximum bandwidth. In the experiments, we make $a_i$ equal to *the revenue difference of one bandwidth interval of class-i* multiplied by 20 seconds. For example, for the class 3, the bandwidth gap is 48, and thus the revenue difference for that gap is 48 in the case of linear revenue function. As a result, $a_3 = 48 * 20 = 960$. This implies that, if the bandwidth of a call is upgraded by one bandwidth level and is changed before 20 seconds, the resulting (net) revenue is negative although the bandwidth of the call increases. Table 2 also shows the adaptation cost for each class which is used for both linear and convex revenue functions.

### 5.3 Numerical Results

Simulation experiments are carried out as the Erlang load of every class increases. Throughout, the Erlang load and the threshold $t_i$ for each class is equally set to provide the same grade of service (e.g., call blocking probability). In the following experiments, $t_i = 5$, which allows the fairly small cell overload probability. Moreover, the handoff call arrival rate is assumed to be proportional

to the new call arrival rate by $h_i = \alpha\lambda_i$ for every class. If $\alpha$ is large, it implies a micro-cellular network environment; otherwise, it can be regarded as simulating macro-cellular networks. We assume $\alpha = 0.5$. Throughout the experiments, the call holding time (CHT) and the cell residence time (CRT) are assumed to follow exponential distribution with mean 500 seconds and 100 seconds, respectively [20]. Also, $C = 1000$ channels leading to a total of 8 Mbps bandwidth. Note that each class-i call will occupy $b_i$ channels. Accordingly, the effective total Erlang Load in a cell will be $\sum_i \lambda_i/\mu_i * b_i$. In the legend, the Erlang load means only $\lambda_i/\mu_i$.

### 5.3.1 Linear Revenue Function

In this section, we evaluate our BAAs in the case of the linear revenue function. Figure 6 shows the *revenue ratio* of three BAAs, which is defined as the ratio of the total revenue of a BAA to that of the BAA-O. When the Erlang Load is 1, there is no overload situation, which results in no difference in the total revenue of BAAs. However, as the Erlang load increases, the revenue difference between BAAs increases accordingly. Note that the revenue of the BAA-RA (with $d = 2$) is fairly close to thta of the BAA-O.

The adaptation costs of three BAAs are shown in Figure 7. Here, as we expected, the BAA-RF shows much higher adaptation cost, which proves our belief that there is a conflicting relation between fairness and anti-adaptation. This reasoning is also verified in Figure 8 where the BAA-A shows the worst fairness deviation. (Plotting of inter-class fairness shows a similar trend.)

Figure 9 shows the total revenue when the adaptation cost increases. Here, we increase the adaptation cost by multiplying *revenue difference of one bandwidth gap* by 10, 20, and 30 seconds, respectively. The values along $x$-axis represent these time intervals in seconds. Also, Erlang load is set to 9. As the adaptation cost increases, the total revenues of the BAA-RA and the BAA-RF decreases notably, while that of the BAA-A shows a minor change. This phenomenon indicates that, when the adaptation cost is large and the cell overload probability cannot be ignored, the adaptation cost can play a dominant role in the total revenue. Accordingly, the CAC algorithm should bound the cell overload probability. Otherwise, we should devise another BAA that pays more attention to the adaptation cost (the BAA-A is an extreme case).

### 5.3.2 Convex Revenue Function

In this section, the numerical results in the case of the convex revenue function are examined. The results of simulation experiments of the convex revenue function show a very similar pattern to those of the linear revenue function and therefore only two plottings are provided here.

Figures 10 and 11 show the adaptation cost and the intra-class fairness of three BAAs. In Figure 10, BAA-A shows the smallest adaptation cost while BAA-RF shows the largest one, which indicates that there is a conflicting relation between the adaptation cost and the fairness. Figure 11 shows the fairness deviation of BAAs and also verifies the above reasoning.

## 6 Conclusions

It is anticipated that demands for adaptive multimedia services will grow in future wireless/mobile networks, especially considering the highly fluctuating bandwidth resources. In an adaptive multimedia framework, bandwidth adaptation algorithms (BAAs) that can satisfy diverse requirements is a challenging issue. We identify the possible objectives of BAAs and propose two algorithms: (i) the BAA for revenue and anti-adaptation (BAA-RA), and (ii) the BAA for revenue and fairness (BAA-RF). Simulations are conducted to compare these algorithms with other BAAs in terms of the total revenue, anti-adaptation, and fairness. The total revenue of the BAA-RA is shown to be fairly close to that of BAA-O (BAA for optimal total revenue). Whereas, the BAA-RF achieves the best total revenue while satisfying the fairness criterion as defined in this paper. Overall, the measured performance metrics of the proposed BAAs show wider differences in the convex revenue function than in the linear one. Simulations reveal that there is a conflicting relation between anti-adaptation and fairness. The analytic model of a cell is also presented in the case of upper-limit call admission control policy.

### Acknowledgments

The work of S. K. Das was parially supported by Texas Advanced Research Program grant TARP-97-35094-013 and Nortel Networks. This work started when T. Kwon was visiting the Center for Research in Wireless Mobility and Networking (CReWMaN) in 1999.

## References


[1] M. Naghshineh and M. Willebeek-LeMair, "End-to-End QoS Provisioning in Multimedia Wireless/Mobile Networks Using an Adaptive Framework," *IEEE Communications Magazine*, Vol. 35, No. 11, pp. 72-81, Nov. 1997.

[2] V. Bharghavan, K. Lee, S. Lu, S. Ha, J. Li, and D. Dwyer, "The TIMELY Adaptive Resource Management Architecture." *IEEE Personal Communications Magazine*, Vol. 5, No. 4, pp. 20-31, Aug. 1998.

[3] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "Rate Adaptation Schemes in Networks with Mobile Hosts," in *Proc. of ACM/IEEE MobiCom '98*, pp. 169-180, Oct. 1998.

[4] S. K. Das and S. K. Sen, "Quality-of-Service Degradation Strategies in Multimedia Wireless Networks," in *Proc. of IEEE Vehicular Technology Conference (VTC '98)*, pp. 1884-1888, Ottawa, May 1998.

[5] K. Lee, "Adaptive Network Support for Mobile Multimedia," in *Proc. of ACM MobiCom '95*, pp. 62-74, 1995.

[6] S. Chakrabarti and R. Wang, "Adaptive Control for Packet Video," in *Proc. of IEEE International Conference on Multimedia Computing and Systems*, pp. 56-62, May 1994.

[7] N. Duffield, K. Ramakrishnan, and A. Reibman, "SAVE: An Algorithm for Smoothed Adaptive Video Over Explicit Rate Networks," *IEEE/ACM Transaction on Networking*, Vol. 6, No. 6, pp. 717-728, Dec. 1998.

[8] S. McCanne, M. Vetterli, and V. Jacobson, "Low-Complexity Video Coding for Receiver-Driven Layered Multicast," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 6, pp. 983-1001, Aug. 1997.

[9] J. Hartung, A. Jacquin, J. Pawlyk, and K. Shipley, "A Real-time Scalable Video Codec for Collaborative Applications over Packet Networks," in *Proc. of ACM Multimedia '98*, pp. 419-426, Bristol, Sept. 1998.

[10] T. Shanableh, M. Ghanbari, "MPEG to H.263 video transcoder," in *Proc. of Packet Video '99*, New York, Apr. 1999.

[11] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*, Wiley-Interscience: New York, NY, 1995.

[12] T. Kwon, Y. Choi, C. Bisdikian, and M. Naghshineh, "Call Admission Control for Adaptive Multimedia in Wireless/Mobile Networks," in *Proc. of ACM Workshop on Wireless Mobile Multimedia (WoWMoM '98)*, pp. 111-116, Dallas, Oct. 1998.

[13] Nicholas Yeadon, "Filters: QoS Support Mechanisms for Multipeer Communications," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, pp. 1245-1262, Sept. 1996.

[14] J. Gomez, A. Campbell, and H. Morikawa, "A Systems Approach to Prediction, Compensation and Adaptation in Wireless Packet Networks," in *Proc. of ACM/IEEE International Workshop on Wireless and Mobile Multimedia* (WoWMoM '98), pp. 92-100, Dallas, Oct. 1998.

[15] C. Chao and W. Chen, "Connection Admission Control for Mobile Multiple-class Personal Communication Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 8, pp. 1618-1626, 1997.

[16] T. Kwon, J. Choi, Y. Choi, and S. K. Das, "Near Optimal Bandwidth Adaptation Algorithm for Adaptive Multimedia Services in Wireless/Mobile Networks," in *Proc. of IEEE VTC '99 Fall*, pp. 874-878, Amsterdam, Sept. 1999.

[17] S. Biswas and B. Sengupta, "Call Admissibility for Multirate Traffic in Wireless ATM Networks," in Proc. of IEEE INFOCOM '97, pp. 650-659, Kobe, Apr. 1997.

[18] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer: London, UK, 1995.

[19] E. Horowitz, S. Sahni, and S. Rajasekaran, *Computer Algorithms/C++*, Computer Science Press: New York, NY, 1996.

[20] Mahmoud Naghshineh and Mischa Schwarz, "Distributed Call Admission Control in Mobile/Wireless Networks," IEEE PIMRC '95, pp. 289-293, 1995.

[21] M. MacDougall, *Simulating computer systems: techniques and tools*, MIT press, 1987.

## Appendix: Analysis of Upper-Limit CAC

Let us now analyze the performance of the UL CAC algorithm. First of all, the UL CAC algorithm determines the space of states. With the help of the product-form solution, the steady probability for state $X$ is given by

$$\pi(X) = G^{-1} \prod_{i=1}^{K} p(x_i) \tag{9}$$

where $G$ is a normalization constant given by

$$G = \sum_{b_{min} \cdot X \leq C} \prod_{i=1}^{K} p(x_i) \tag{10}$$

Here $b_{min} = (b_{1,1}, b_{2,1}, ..., b_{K,1})$ is a vector of minimum bandwidth values of each class. Moreover, $p(x_i)$ means the probability intensity that there are $x_i$ number of class-$i$ calls in state $X$ and is given by

$$p(x_i) = \begin{cases} \left(\frac{\lambda+h_i}{\mu_i+h}\right)^{x_i} / x_i! & \text{if } x_i \leq t_i \\ \left(\frac{\lambda+h_i}{\mu_i+h}\right)^{t_i} \left(\frac{h_i}{\mu_i+h}\right)^{x_i - t_i} / x_i! & \text{if } x_i > t_i \end{cases} \tag{11}$$

With these steady-state probabilities, the call blocking probability is given by Eq. (12).

$$P_{B_i} = \sum_{x_i \geq t_i} \pi(X) \tag{12}$$

where $e_i$ is a vector of zeros, except for a one for the $i$-th component. Table 3 shows the call blocking probability calculated by Eq. (12) and obtained from the simulation experiments. Note that the call blocking probability calculated by analysis is independent of the class because the threshold for each class is set equally. Also, the handoff dropping probability is negligible, so its plotting is omitted. Finally, the cell overload probability, $P_O$, is given by

$$P_O = \sum_{b_{max} \cdot X > C} \pi(X) \tag{13}$$

Table 4 and 5 show the cell overload probability, $P_O$, versus Erlang load when $t_i = 5$ and 7, respectively. In both cases, the Erlang load of each class increases equally, and $t_i$ of each class is set equally.

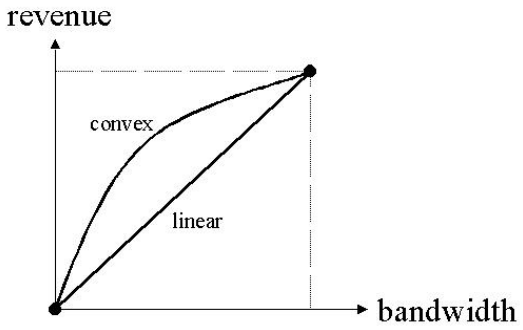



Figure 1: Examples of revenue function

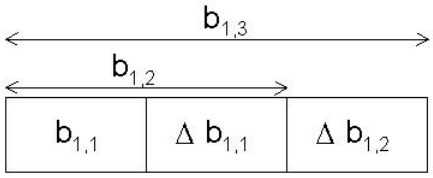



Figure 2: Example of a Class-1 Adaptive Multimedia Stream

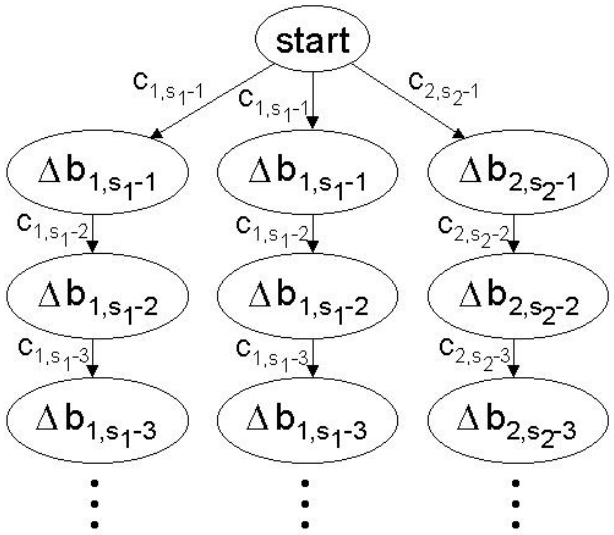



Figure 3: Graph for BAA-R

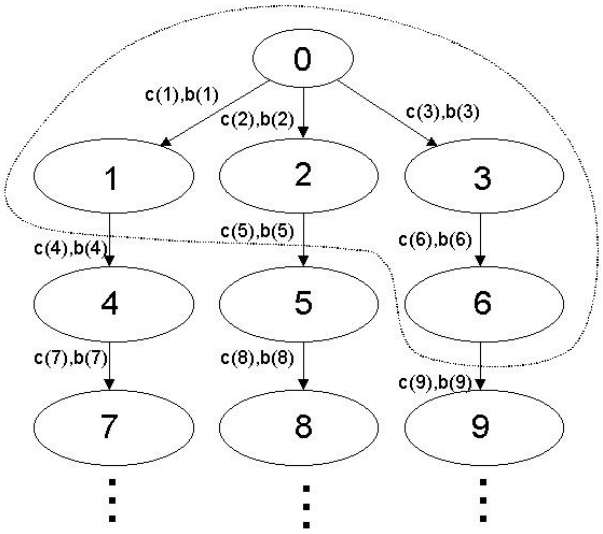
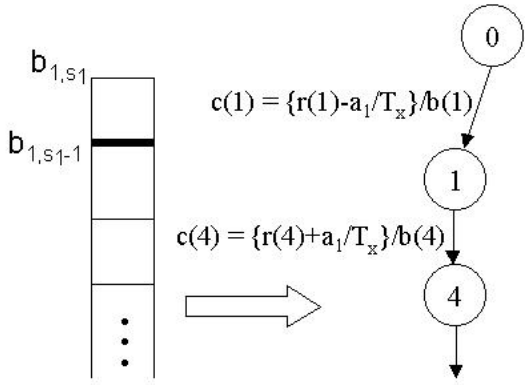
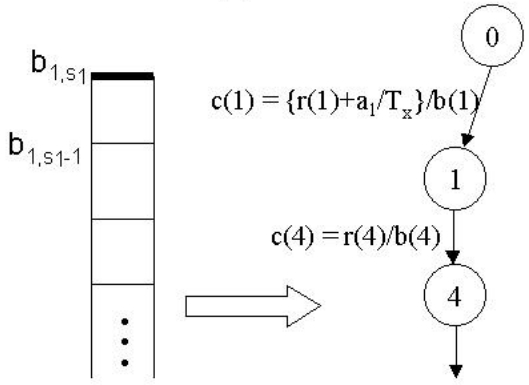



Figure 4: Modified Graph Representation




(a) 1st case




(b) 2nd case

Figure 5: Incorporation of Anti-Adaptation

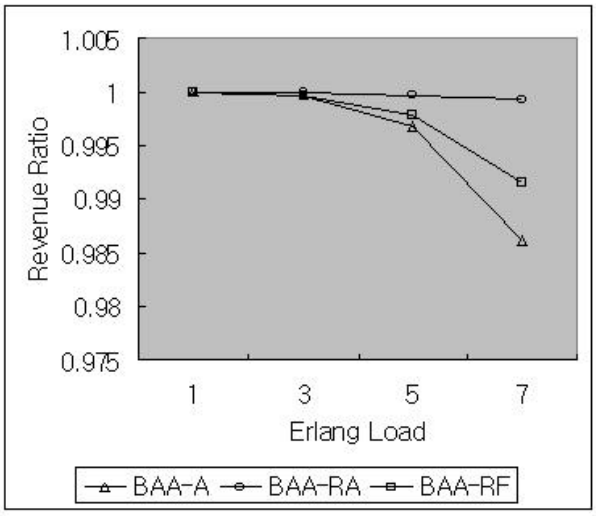



Figure 6: Revenue Ratio vs. Erlang Load (Linear)

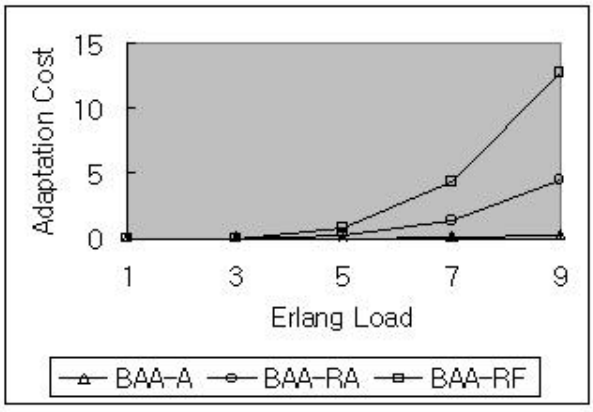



Figure 7: Adaptation Cost vs. Erlang Load (Linear)

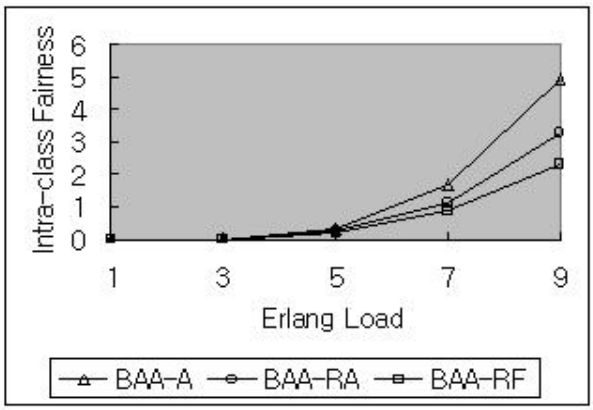



Figure 8: Intra-class Fairness vs. Erlang Load (Linear)

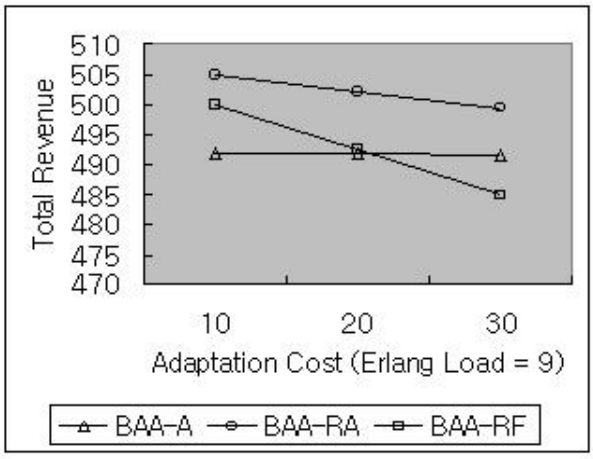



Figure 9: Total Revenue vs. Adaptation Cost (Linear)

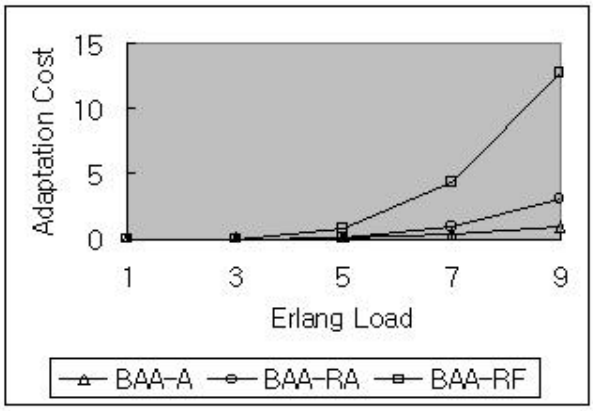



Figure 10: Adaptation Cost vs. Erlang Load (Convex)

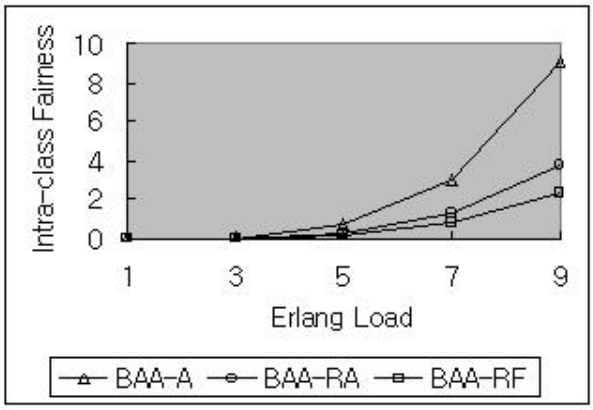



Figure 11: Intra-class Fairness vs. Erlang Load (Convex)

Table 1: BAA for Revenue (BAA-R)

```
1   N ← 0, sum_b ← 0
2   b_req ← b_max · X − C − sum_b
3   while(b_req > Σ^d U(N)){
4       choose the minimum cost edge in U(N)
5       add n (the node of this edge) to N
6       sum_b ← sum_b + b(n)
7       b_req ← b_max · x − C − sum_b
8   }
9   R_min ← ∞
10  for(m is 1 to d){
11      for( every possible m edges in U(N)){
12          S ← selected m edges
13          if(B(S) ≥ b_req and R(S) < R_min){
14              S_min ← S
15              R_min ← R(S)
16          }
17      }
18  }
```

Table 2: Bandwidth Values (in channels)

| Services | $s_i$ | $V_i$ | $a_i$ |
|---|---|---|---|
| Class 1 | 4 | {2, 4, 6, 8} | 40 |
| Class 2 | 4 | {8, 16, 24, 32} | 160 |
| Class 3 | 4 | {48, 96, 144, 192} | 960 |

Table 3: Call Blocking Probability ($t_i = 5$)

| Load | analysis | simulation | | |
|---|---|---|---|---|
| | | class 1 | class 2 | class 3 |
| 3 | 0.000975 | 0.000951 | 0.000966 | 0.001183 |
| 5 | 0.007834 | 0.007349 | 0.008147 | 0.007839 |
| 7 | 0.026461 | 0.026332 | 0.025593 | 0.025884 |
| 9 | 0.058923 | 0.060115 | 0.058022 | 0.058399 |
| 11 | 0.103241 | 0.105412 | 0.100334 | 0.102873 |
| 13 | 0.155504 | 0.160023 | 0.158060 | 0.153816 |
| 15 | 0.211738 | 0.209334 | 0.207461 | 0.206327 |

Table 4: Cell Overload Probability ($t_i = 5$)

| Load | analysis | simulation |
|---|---|---|
| 3 | 0.000019 | 0.000021 |
| 5 | 0.000403 | 0.000395 |
| 7 | 0.002485 | 0.002518 |
| 9 | 0.008333 | 0.008292 |
| 11 | 0.019605 | 0.019646 |
| 13 | 0.036793 | 0.036962 |
| 15 | 0.059276 | 0.059333 |

Table 5: Cell Overload Probability ($t_i = 7$)

| Load | analysis | simulation |
|---|---|---|
| 3 | 0.000033 | 0.000036 |
| 5 | 0.000629 | 0.000604 |
| 7 | 0.003763 | 0.003704 |
| 9 | 0.012650 | 0.012669 |
| 11 | 0.030315 | 0.030496 |
| 13 | 0.058386 | 0.058376 |
| 15 | 0.096807 | 0.095825 |